

Windows XP SP2

Robert C. Seacord, Software Engineering Institute [vita¹]

Copyright © 2005, 2008 Pearson Education, Inc.

2005-09-27; Updated 2008-10-06

L4 / D/P, L²

Different versions of the Windows operating systems contain different implementations of the heap. The Windows XP SP2 release has two significant improvements over earlier heap implementations that make it more difficult to exploit.

Development Context

Dynamic memory management

Technology Context

C++, C, Win32

Attacks

Attacker executes arbitrary code on machine with permissions of compromised process or changes the behavior of the program.

Risk

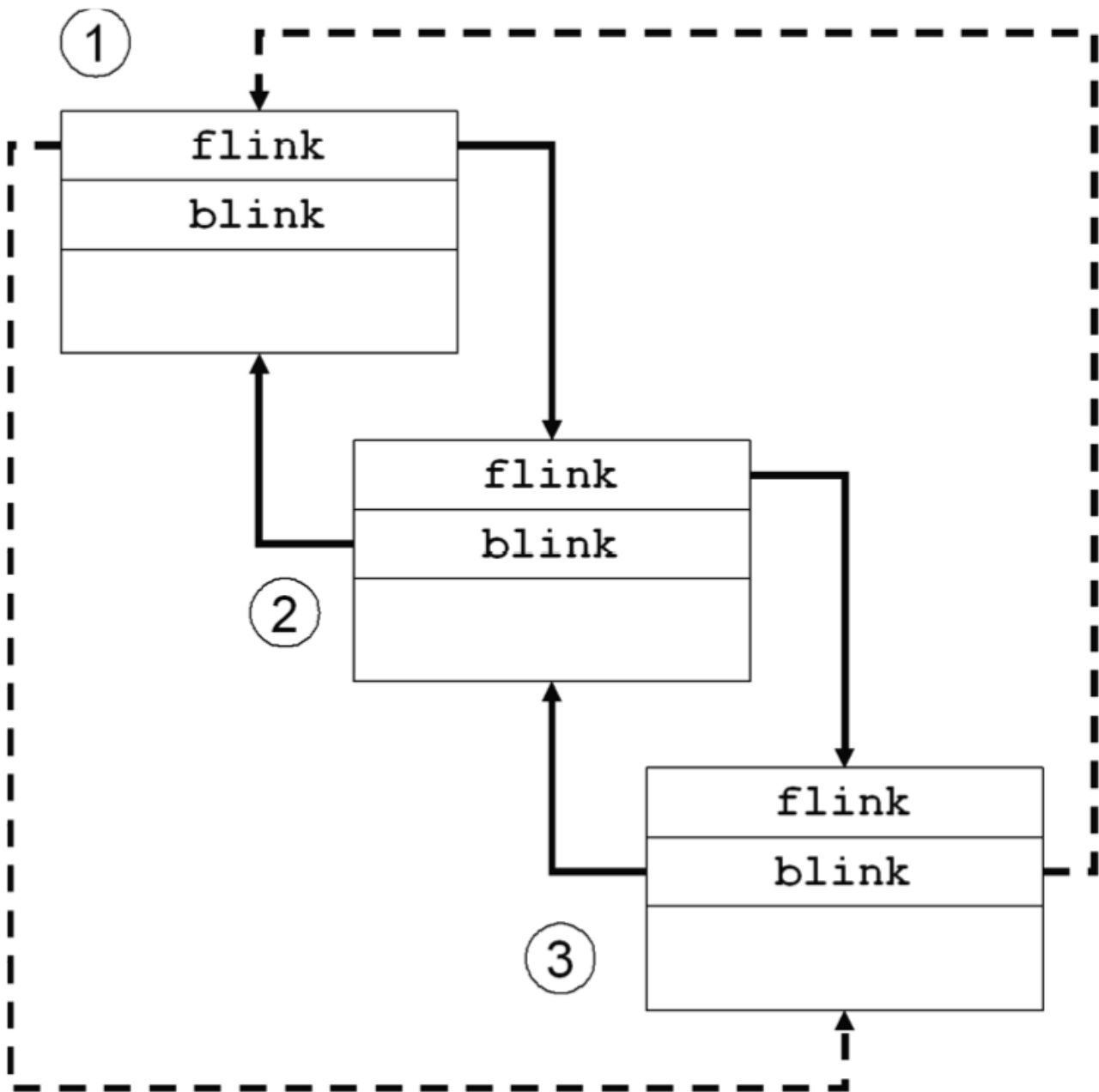
Standard C dynamic memory management functions such as `malloc()`, `calloc()`, `realloc()`, and `free()` [ISO/IEC 99] are prone to programmer mistakes that can lead to vulnerabilities resulting from buffer overflow in the heap, writing to already freed memory, and freeing the same memory multiple times (e.g., double-free vulnerabilities).

Description

Different versions of the Windows operating systems contain different implementations of the heap. The Windows XP SP2 release has two significant improvements over earlier heap implementations that make it more difficult to exploit.

Figure 1. Free list management data structures

1. http://buildsecurityin.us-cert.gov/bsi/about_us/authors/274-BSI.html (Seacord, Robert C.)



The first improvement is an eight-bit canary at the end of each chunk. The size of the canary was selected for performance reasons, particularly in systems that allocate and free large numbers of relatively small memory chunks. Unfortunately, because there are only 256 unique values, this canary can be easily brute-forced in situations where the application under attack behaves in a highly deterministic fashion.

The second improvement is additional checking in the free list management code as shown in Figure 1, which includes three unallocated chunks of memory in a doubly linked list. When chunk 2 is removed from the linked list, the forward pointer of the previous chunk (chunk 1) is updated to point to chunk 3 and the backward pointer of the following chunk (chunk 3) is updated to link back to chunk 1. Many heap exploits work by modifying the values for flink and blink in chunk 2. In XP SP2, the memory manager does not update the pointers in chunk 1 before verifying that the existing pointers reference valid memory chunks.

Although this change improves the free list management code of the heap, it does not address the look-aside list management code. RtlHeap maintains a doubly linked free list and a singly linked look-aside list. Windows XP SP2 does not include header integrity checks for the look-aside list. Alexander Anisimov has published a detailed description of how this security flaw can be exploited to execute arbitrary code [Anisimov 2005].

Unfortunately, restricting deployment of an application to Windows XP SP2 systems is not always an option because most commercial applications are developed for a broad range of platforms, not a specific service pack of a particular operating system. However, developing and testing under XP SP2 may help to identify and correct coding problems before deploying to other platforms.

References

- | | |
|-----------------|---|
| [Anisimov 2005] | Anisimov, Alexander. <i>Defeating Microsoft Windows XP SP2 Heap Protection and DEP Bypass</i> ¹⁸ (2005). |
| [ISO/IEC 99] | ISO/IEC. <i>ISO/IEC 9899 Second edition 1999-12-01 Programming Language — C</i> . International Organization for Standardization, 1999. |

Pearson Education, Inc. Copyright

This material is excerpted from *Secure Coding in C and C++*, by Robert C. Seacord, copyright © 2006 by Pearson Education, Inc., published as a CERT[®] book in the SEI Series in Software Engineering. All rights reserved. It is reprinted with permission and may not be further reproduced or distributed without the prior written consent of Pearson Education, Inc.